

SFA Modernization Partner
United States Department of Education
Student Financial Assistance



ITA Release 1.0
Strategic Assessment

Task Order #46
Deliverable # 46.1.1

Version 1.0

May 21, 2001



Table of Contents

1	EXECUTIVE SUMMARY.....	8
1.1	INTRODUCTION/OBJECTIVES	8
1.2	DESCRIPTION OF SECTIONS.....	8
1.3	APPROACH.....	8
1.4	ASSUMPTIONS.....	9
1.5	ITA 1.0 OVERVIEW.....	9
1.6	LIST OF PROJECTS.....	9
1.7	EXISTING CAPABILITIES.....	10
1.8	ENHANCEMENT AREAS	10
1.9	ITA 1.0 UPGRADES	11
1.10	ITA 2.0 SERVICE OFFERINGS	11
2	ITA ARCHITECTURE	13
2.1	ITA OVERVIEW	13
2.2	ITA PRODUCTS AND UPGRADES	14
2.3	ITA RELEASE 1.0	14
2.3.1	DESCRIPTION.....	14
2.3.2	DEVIATIONS.....	14
2.4	ITA RELEASE 2.0	16
2.5	APPROACH FOR BUILDING NEW SERVICES	17



2.6	NEW SERVICES 2.0.....	17
2.6.1	COMPONENT FACTORY	18
2.6.2	CONFIGURATION.....	18
2.6.3	EXCEPTION HANDLING	19
2.6.4	FILE TRANSFER PROTOCOL	19
2.6.5	LOGGING.....	20
2.6.6	MAIL	20
2.6.7	PERSISTENCE	21
2.6.8	SEARCH.....	21
2.6.9	SECURITY.....	22
2.6.10	SERVLET	22
2.6.11	SESSION MANAGEMENT.....	22
2.6.12	TIMEOUT.....	23
3	SFA PROJECTS.....	24
3.1	OVERVIEW	24
3.2	CBS2	24
3.2.1	FUNCTIONAL DESCRIPTION.....	24
3.2.2	TECHNICAL DESCRIPTION	25
3.2.3	STATUS.....	26
3.2.4	ITA 1.0 SERVICES USED	26
3.2.5	ENHANCEMENTS/UPGRADES.....	26
3.2.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	26



3.2.7	SCHEDULE.....	27
3.3	CFO.....	28
3.3.1	FUNCTIONAL DESCRIPTION.....	28
3.3.2	TECHNICAL DESCRIPTION.....	29
3.3.3	STATUS.....	30
3.3.4	ITA 1.0 SERVICES USED	30
3.3.5	ENHANCEMENTS/UPGRADES.....	30
3.3.6	POTENTIAL ITA 2.0 SERVICES.....	30
3.3.7	SCHEDULE.....	30
3.4	COD.....	31
3.4.1	FUNCTIONAL DESCRIPTION.....	31
3.4.2	TECHNICAL DESCRIPTION.....	31
3.4.3	STATUS.....	32
3.4.4	ITA 1.0 SERVICES USED	32
3.4.5	ENHANCEMENTS/UPGRADES.....	32
3.4.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	32
3.4.7	SCHEDULE.....	32
3.5	DLM.....	32
3.5.1	FUNCTIONAL DESCRIPTION.....	32
3.5.2	TECHNICAL DESCRIPTION.....	34
3.5.3	STATUS.....	34
3.5.4	ITA 1.0 SERVICES USED	34



3.5.5	ENHANCEMENTS/UPGRADES.....	34
3.5.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	34
3.5.7	SCHEDULE.....	35
3.6	EAI.....	35
3.6.1	FUNCTIONAL DESCRIPTION.....	35
3.6.2	TECHNICAL DESCRIPTION.....	36
3.6.3	STATUS.....	36
3.6.4	ITA 1.0 SERVICES USED	36
3.6.5	ENHANCEMENTS/UPGRADES.....	36
3.6.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	37
3.6.7	SCHEDULE.....	37
3.7	EIP.....	37
3.7.1	FUNCTIONAL DESCRIPTION.....	37
3.7.2	TECHNICAL DESCRIPTION.....	37
3.7.3	STATUS.....	39
3.7.4	ITA 1.0 SERVICES USED	39
3.7.5	ENHANCEMENTS/UPGRADES.....	39
3.7.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	39
3.7.7	SCHEDULE.....	41
3.8	FAFSA	41
3.8.1	FUNCTIONAL DESCRIPTION.....	41
3.8.2	TECHNICAL DESCRIPTION.....	42



3.8.3	STATUS.....	43
3.8.4	ITA 1.0 SERVICES USED	43
3.8.5	ENHANCEMENTS/UPGRADES.....	43
3.8.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	43
3.8.7	SCHEDULE.....	44
3.9	FP DATA MART.....	44
3.9.1	FUNCTIONAL DESCRIPTION.....	44
3.9.2	TECHNICAL DESCRIPTION.....	45
3.9.3	STATUS.....	47
3.9.4	ITA 1.0 SERVICES USED	47
3.9.5	ENHANCEMENTS/UPGRADES.....	47
3.9.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	47
3.9.7	SCHEDULE.....	47
3.10	IFAP	48
3.10.1	FUNCTIONAL DESCRIPTION.....	48
3.10.2	TECHNICAL DESCRIPTION.....	49
3.10.3	STATUS.....	49
3.10.4	ITA 1.0 SERVICES USED	49
3.10.5	ENHANCEMENTS/UPGRADES.....	50
3.10.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	50
3.10.7	SCHEDULE.....	50
3.11	INTRANET (SFANet) 2.0	50



3.11.1	FUNCTIONAL DESCRIPTION.....	50
3.11.2	TECHNICAL DESCRIPTION.....	51
3.11.3	STATUS.....	53
3.11.4	ITA 1.0 SERVICES USED.....	53
3.11.5	ENHANCEMENTS/UPGRADES.....	53
3.11.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	53
3.11.7	SCHEDULE.....	54
3.12	SCHOOLS PORTAL	54
3.12.1	FUNCTIONAL DESCRIPTION.....	54
3.12.2	TECHNICAL DESCRIPTION.....	54
3.12.3	STATUS.....	55
3.12.4	ITA 1.0 SERVICES USED.....	55
3.12.5	ENHANCEMENTS/UPGRADES.....	55
3.12.6	CURRENT/POTENTIAL ITA 2.0 SERVICES.....	55
3.12.7	SCHEDULE.....	55
4	ASSESSMENT SUMMARY.....	56
4.1	STRENGTHS	56
4.1.1	STANDARD PRODUCT FRAMEWORK.....	56
4.1.2	OPERATIONS SUPPORT.....	56
4.1.3	BUILD/SETUP ENVIRONMENTS.....	56
4.1.4	SME SUPPORT.....	56
4.1.5	TROUBLESHOOTING AND BUG FIXES.....	56



4.2	ENHANCEMENT AREAS	57
4.2.1	COMMUNICATION.....	57
4.2.2	USE OF ITA CORE SERVICES.....	57
4.2.3	PRODUCTION HARDWARE IMPLEMENTATION.....	57
4.2.4	ISSUE TRACKING.....	57
4.2.5	CONFIGURATION MANAGEMENT.....	58
4.2.6	ITA R1.0 SERVICES TOO GENERAL.....	58
4.2.7	EXCEPTION HANDLING	58
4.2.8	LOGGING.....	58
4.2.9	DATABASE.....	58
4.2.10	MAIL.....	58
4.2.11	PROCESS MONITORING.....	59
4.2.12	LACK OF SOFTWARE FUNDING.....	59
4.2.13	SECURITY.....	59
4.3	PRIORITIZED RECOMMENDATIONS	60
4.3.1	COMMON SERVICES.....	62
APPENDIX A – APPLICATION COMMON SERVICES MATRIX.....		63
APPENDIX B – INTEGRATED TECHNICAL ARCHITECTURE PRODUCT STANDARDS		64



1 Executive Summary

1.1 Introduction/Objectives

The *Integrated Technical Architecture (ITA) Release 1.0 (R1.0) Strategic Assessment* analyzes the ITA R1.0 architecture and identifies the strengths and improvement areas for the deployed architecture. This analysis provides a thorough review of all of the SFA applications that are deployed with or are planning to use the ITA architecture.

Based on the analysis and input from the application teams, the document also identifies and prioritizes ITA enhancement areas for the ITA Release 2.0 (R2.0) architecture.

This document provides a summary of the Student Financial Assistance's (SFA) projects, how they have implemented ITA R1.0, strengths, areas for improvement, prioritized enhancement requests, and new services for ITA R2.0.

1.2 Description of Sections

The ITA R1.0 Strategic Assessment is divided into the following sections:

- Section 1 provides a high level overview of the strategic assessment, including the projects that were assessed, an overview of ITA R1.0, strengths and improvement areas of ITA R1.0, enhancements, and new services that will be provided in ITA R2.0.
- Section 2 provides a detailed overview of the ITA R1.0 architecture, products, enhancements, upgrades, and new common services for ITA R2.0.
- Section 3 provides an analysis of the SFA applications.
- Section 4 provides an assessment summary with strengths, improvement areas, and prioritized recommendations.
- Appendix A lists the new ITA R2.0 Java common services that SFA applications may implement.

1.3 Approach

The ITA team collected information for the Strategic Assessment through interviews with application team leads and members of the ITA R1.0 team. Additional information on SFA's projects and ITA R1.0 were collected through the review of existing application documentation.



The ITA team asked the application leads a common set of questions regarding the applications' functional and technical descriptions, ITA R1.0 services used, strengths and improvement areas of ITA R1.0, and potential ITA R2.0 services.

1.4 Assumptions

- The projects listed in section 1.6 are a complete list of SFA's projects that implemented ITA R1.0 or plan to implement ITA R2.0.
- The application teams have provided application volume estimates. In order to capacity plan or size hardware for the applications, these volumes will need to be validated against current and projected volumes.

1.5 ITA 1.0 Overview

ITA R1.0 provided SFA with the core functionality to support the deployment of the initial SFA applications. ITA R1.0 provided a standardized, reusable infrastructure for enabling business capabilities. Future releases will build on R1.0, adding new technologies, as required.

The ITA framework consists of four domains:

- Internet Domain
- Enterprise Application Integration Domain
- Enterprise Data Domain
- Legacy Domain

For R1.0, ITA supported the Internet and Enterprise Data Domains.

Based on schedule constraints during R1.0, deviations in the implementation of the ITA architecture were deployed to maintain schedule and to provide functionality to support the deployed applications. Section four of the Strategic Assessment addresses these deviations and provides a plan for bringing the architecture and services into the scope of the enterprise ITA design for R2.0.

1.6 List of Projects

This document will assess the usage of ITA by the following SFA projects:

- Campus Based-System 2 (CBS2)



- CFO Data Mart
- Common Origination and Disbursement (COD)
- Delinquency Loan Mart (DLM)
- Enterprise Application Integration (EAI)
- Enterprise Information Portal (EIP)
- Free Application for Federal Student Aid (FAFSA)
- Financial Partner's (FP) Data Mart
- Information for Financial Aid Professionals (IFAP)
- Intranet 2.0 (SFANet)
- Schools Portal

1.7 Existing Capabilities

ITA R1.0 provided the following support and capabilities for SFA's applications:

- Standard product framework
- Build and setup of environments
- Subject Matter Expert (SME) support
- Troubleshooting and bug fixes

1.8 Enhancement Areas

The key enhancement areas identified in ITA R1.0 are:

- Communication
- Use of ITA core services
- Production hardware implementation
- Issue tracking
- Configuration management
- Additional ITA services
- Exception handling
- Logging



- Database
- Mail
- Process monitoring
- Software funding
- Security

1.9 ITA 1.0 Upgrades

Based on the analysis of ITA R1.0, the following enhancements have been identified:

- Upgrade software
 - WebSphere Application Server (WAS)
 - Web Server (IHS)
 - NetworkDispatcher
 - Autonomy
 - Interwoven
- Implement configuration management process

1.10 ITA 2.0 Service Offerings

ITA R2.0 will provide SFA application teams with the following new services:

- Coding Standards (Java & SQL)
- Technical Architecture Design Reviews
- Common Services
 - Component Factory
 - Configuration Framework
 - Exception Handling Framework
 - File Transfer Protocol Framework
 - Logging Framework
 - Mail Framework
 - Persistence Framework



- Search Framework
- Security Framework
- Servlet Framework
- Session Framework
- Timeout Framework



2 ITA Architecture

2.1 ITA Overview

The ITA provides an infrastructure that facilitates the integration of disparate data resources, business services, and process automation. The ITA is flexible, easily modified and provides for reuse as the following changes to SFA's enterprise computing environment are introduced:

- Shift in business processes
- Additional data resources
- Future technology shifts
- A move to provide self-service access to ITA

The ITA is based on four core architecture domains within the SFA technical environment, which are targeted at reducing stovepipe systems, islands of technology, and the need for customized point-to-point system interfaces. The four new domains include the

- Internet Domain
- Enterprise Application Integration Domain
- Enterprise Data Domain
- Legacy Domain

Together the four core ITA domains provide the necessary infrastructure to implement a service-oriented architecture. However, in order to provide this infrastructure each domain must provide support for seamless integration between domain touch-points. Each domain must provide a set of interfaces or connectors that allow integration with the services of intersecting domains. For example, the Internet Domain must provide interfaces to the Data Domain in order to provide persistence for stateful business objects. The Data Domain must provide adapters to disparate data sources in order to feed the extract/transform/load process. The Enterprise Architecture Integration (EAI) Domain must provide connectivity to different legacy architectures and data sources in order to provide integration with existing SFA systems.

The four domains overlap to provide seamless integration between domain services. The EAI layer is the glue for providing the integration between the domains where interfaces do not exist. The EAI layer provides a set of application adapters and communication services that can span domains thereby providing additional integration points between domains.



2.2 ITA Products and Upgrades

One of the main goals of the ITA is to standardize hardware and software products across SFA. In order to ensure that the standard framework is current the ITA will periodically upgrade the version of software supported. Appendix B: Integrated Technical Architecture Product Standards summarizes the products and the versions supported by ITA 1.0 and ITA 2.0.

2.3 ITA Release 1.0

2.3.1 Description

ITA R1.0 deployed the Internet and Enterprise Data Domain and provided SFA's application teams with a core set of products and SME support. ITA built environments, provided product troubleshooting, debugged code, and provided support for the products and application code.

2.3.2 Deviations

The following deviations from the planned ITA architecture were implemented during Release 1.0.

- Failover & Clustering
- Use of CGI for Autonomy search capability.
- Use of IP addresses, not URLs, for sites in Autonomy spider.
- Development, test, and staging all co-located on one server.
- No change management system in place.
- Application Java code may not comply with SFA and ITA Coding Standards
- Implementation of JRun as Application Server in Schools Portal

Failover & Clustering

ITA R1.0 did not support clustering for WebSphere Application Server. As a result, there was neither load balancing nor failover of the boxes. The lack of clustering has been addressed in the ITA R2.0 architecture. The ITA has upgraded all new development efforts to WAS 3.5, which provides configurable clustering out of the box. All of the ITA R2.0 development, test, and production environments will now be clustered and support load balancing and failover. The WAS boxes will share a common repository to support the clustering and will also support



session affinity between the app & web server. Load balancing has been set up in a round robin configuration.

Autonomy - Use of CGI for search functionality in Web browser

In ITA R1.0, several applications used CGI to implement Autonomy's search functionality. The IFAP application created custom CGI programs using the Autonomy Knowledge Builder API and the C programming language. The Schools Portal application used Autonomy provided CGI programs and template HTML files. The Intranet R2.0 application created custom CGI programs using the Autonomy Knowledge Builder API and the C programming language.

CGI is not scalable, does not separate presentation and business logic, and is slower than other technologies. CGI also presents security concerns.

An autonomy upgrade will provide Java support in the new release of the Autonomy API. Based on these new APIs, the search capability for these programs may be rewritten using Java. ITA Release 2.0 will provide applications with a common service to make the Autonomy search calls.

Use of IP addresses and not URLs for Autonomy spider Web sites

The Schools Portal application uses the Autonomy spider to gather external content for searching. Since there is no DNS capability to resolve URLs, the spider references IP addresses. If an IP address changes for a web site, the content for that site must be completely rebuilt. IP addresses are not informative and can lead users to unfamiliar Web sites.

For ITA R2.0, SFA should implement DNS capability for Autonomy Servers E7 and E15. After implementing the DNS capability, SFA will need to re-index all of the Schools Portal content.

Development, Test, and Staging environments all on one server

In ITA R1.0, three instances of each product and application are installed on single servers. Installing the development, test, and staging environments on single servers makes migration between environments difficult to control and validate

For ITA R2.0, SFA should move development to separate servers from test and staging.

Change management system was not implemented

In ITA R1.0, a change management system was not implemented. Without a change management system in place, source code control is a manual, user intensive, and unreliable



process. Source code versions and application baselines were not established, so it was difficult to revert to an older version of source code and isolate code problems.

For ITA R2.0, SFA should implement a change management tool and process.

Application Java Code may not follow SFA and ITA Coding Standards

IFAP and Schools Portal source code may not follow SFA and ITA standards. Code that deviates from the SFA and ITA standards may not be supported in WAS 3.5 or future versions.

Future development efforts should follow SFA and ITA Java Coding standards to ensure that the applications conform to the latest standards and are compatible with the core ITA infrastructure.

Implementation of JRun as Application Server in Schools Portal

In ITA R1.0, the Schools Portal used JRun for its application server. JRun was not part of the planned ITA architecture.

In ITA R2.0, in accordance with the ITA standard products, development efforts will use WebSphere Application Server.

2.4 ITA Release 2.0

ITA R2.0 will build on R1.0 in order to provide SFA with a more robust core infrastructure. In addition to providing the application teams with a core set of products and SME support, ITA R2.0 will provide a set of Java 2 Enterprise Edition (J2EE)-compliant common services. These common services have been identified from the strategic assessment of the applications and their current usage of ITA 1.0 as well as any needs that were identified.

ITA R2.0 will provide the application teams with the following support in accordance with the corresponding target percentage of time allocated:

- ITA R1.0 Strategic Assessment (10%)
- ITA Product Analysis, Software Upgrade Strategy, and Software Upgrade Rollout (15%)
- Reusable Component Development (Middleware Component Libraries) (25%)
- Developer Best Practices Guidelines (implementation of key concepts such as data connection pooling and session management) (10%)



- Operational and Hardware Services (including Failover, Clustering, and Systems Recovery Processes) (15%)
- Product Specialist Support (10%)
- ITA Release 2.0 Product Rollout (to the application development teams) (15%)

ITA R2.0 will also provide application teams with support in the following areas: general support for environments, technical architecture design reviews (to ensure that coding standards and best practices are being followed), and capacity planning for performance test and production environments.

2.5 Approach for Building New Services

The ITA team will build new services for ITA R2.0 based on an open technology and J2EE architecture. The ITA team will leverage previous Accenture and industry experience to design and develop a robust set of core common services that specifically address SFA enterprise application requirements.

2.6 New Services 2.0

Based on the ITA R1.0 Strategic Assessment, ITA R2.0 will provide the following new common services:

- Component Factory
- Configuration
- Exception Handling
- File Transfer Protocol
- Logging
- Mail
- Persistence
- Search
- Security
- Servlet
- Session
- Timeout



2.6.1 Component Factory

The component factory defines a general and extensible factory mechanism. The service enables developers to completely decouple how objects and components are instantiated from their use, going beyond simplifying EJB and JNDI access. In addition, the meta-information used to define the object production mechanism and surrounding context is provided via configuration properties, thereby allowing production instances to be changed with minimal effort. The component factory enables the definition of clear migration strategies from one architectural approach to another.

The component factory provides an extensible mechanism to associate producers, targets, and context together to produce objects. A common set of producers will be provided that cover a large percentage of scenarios relating to co-located and J2EE object/component production.

This framework will provide the following features:

- Standard methods of producing objects when parameters are passed
- Common parameters to be passed when producing objects
- Standards for coding and designing objects
- Examples for using the component factory framework

2.6.2 Configuration

Both the architecture and the applications being developed on IBM WebSphere require configurations to be loaded from an external file. In the case of the architecture, the logging framework must load in the initial configuration of the logs. Within the application, properties containing database information must be loaded. Both the architecture and the applications utilize different approaches to reading in the files, making the code complex and non-standard. This function of the configuration file reading can be abstracted into the architecture, providing a standard approach to the reading of all configuration files. This also lessens the impact to the application developers when needing to add an additional parameter to this file by providing a single source from which to obtain the configurations.

The configuration framework will provide a standard for configuration file input. The configuration settings for an application will reside in a single implementation of the configuration manager. The mechanism for reading the configuration files should be abstracted into the architecture layer to avoid any duplicate file reading logic being written with the application code. The configuration framework will also abstract the representation of configuration files (the style in which they are written) so that if the files change formats,



only the configuration framework will require coding changes – no coding changes will need to be made within the application or architecture code.

The configuration framework will load the configuration file upon startup, cache the configuration into memory for easy access, and provide the functionality of searching through the configuration. It also provides the capability of determining the operating system on which it currently resides.

2.6.3 Exception Handling

An exception is a code or language construct that indicates when an unusual or unexpected error condition occurs in an application. Examples of exceptions could be hardware, network, I/O, or memory problems. If an exception is ‘handled’ in code, it can be dealt with gracefully and will not necessarily have to cause program termination. Exception handling provides a mechanism for writing robust, resilient code that is capable of dealing with the unexpected.

The goal of an exception handling framework is to make exception handling standardized and straightforward and always done correctly, thereby eliminating uncaught exception scenarios.

The exception handling framework provides the following general services and components:

- Guidelines for identification and responding to exceptions
- Guidelines for throwing and catching exceptions
- Base exception classes
- Default last-resort exception handlers
- Simple interface for integration of logging exceptions
- A generic exception class that must be thrown by all components in the application. It contains a status code that represents the type of exception. This generic exception can be extended for specific errors
- An exception factory class that will be used to create exceptions, that will automatically log and assign a unique id. This unique id will be displayed to the user if necessary in order to uniquely identify the associated log and therefore all the associated information.

2.6.4 File Transfer Protocol

The File Transfer Protocol (FTP) framework provides the ability to transfer files between different platforms in the SFA environment. The FTP framework is an extension of a standard



file transfer application which provides a java native interface to the "put" and "get" process. Files can be moved from one box to another regardless of platform. Exception handling will be integrated with the framework to ensure that files have been accurately and completely transferred. This framework can be used for online processing within a Java thread of execution or in a batch mode where it would be called from an NT scheduled job or UNIX cron job.

2.6.5 Logging

The logging framework enables users to track and identify the source of errors. The logging framework uses a routine to determine the originating class and method for the logging call. This provides complete and descriptive logs that enable operations personnel to view the logs and quickly determine where the instance or error occurred and possibly what caused it. Logs are not tied to exceptions only; they are customized to record access and other useful information in troubleshooting and analyzing an application.

Users can define handlers to be global or assign them to specific, named loggers. Loggers can be associated to both the global handler set and to specific handlers. The formatting of the message only happens at the handler. Both loggers and handlers can filter messages based on some function, and by the level of the message.

A set of appropriate handlers will be defined for the given application. This could mean Error or higher goes to a specific handler and Info and higher goes to another. In general, these handlers will be global. Application code should log to a named logger appropriate to its subsystem. Finer control of log levels is set within the handler. However, further filtering will require the creation of a custom filter that is attached to the handler.

The logging framework provides the following features:

- Custom logging
- Filtering messages by level
- Integration with the exception handling framework

2.6.6 Mail

The existing ITA 1.0 applications make use of two mail frameworks, one is JavaMail and the other is a database batch product. The database batch mailing is expected to be phased out and ITA 2.0 will add additional features to the existing JavaMail, including performance tuning, integration with logging and exception handling, as well as a developer handbook.



2.6.7 Persistence

The persistence framework encapsulates the behavior needed to make objects persistent. Specifically, a persistence framework reads, writes, and deletes objects to/from permanent storage. The persistence provides full encapsulation of the persistence mechanism. Application developers can send a save, delete, or retrieve message to the persistence framework and the framework will handle the rest of the work with the database.

The persistence framework also provides the ability to implement persistence behavior on multiple objects concurrently. The framework supports saving, deleting, or retrieving many objects at once depending upon a specific criterion.

The persistence layer can implement transactional behavior on objects. A transaction is defined as a combination of actions implemented on several objects concurrently. An example is adding an object to a database and deleting another object from another database and being able to rollback the entire transaction if an error occurred.

The persistence layer uses pooling resources available to help maintain efficient use of the database. If a single client has the ability to request every record from a datasource, then that client may be able to consume almost all resources of that datasource. The persistence framework uses a controlled approach that does not allow runaway use of a resource.

The persistence layer can dynamically run stored procedures on the database or submit SQL directly from the application. The persistence framework includes application supplied data classes that allow the framework to know how the schema of the database it is connected to.

2.6.8 Search

The search framework simplifies, standardizes, and improves the use of the Autonomy search engine. This framework complies with J2EE standards instead of using CGI as in the current search engine interface. The framework consists of a search wrapper class that provides a common way to access the Autonomy API and utilize its features.

The search wrapper implements the following Autonomy features:

- Query search engine
- Display search results
- Suggest additional search results
- Highlight keywords from search results



2.6.9 Security

The security framework provides an interim bridge between applications' current unique security solutions and the future selection of a Web Access Control (WAC) tool. The security framework allows for the authorization and authentication of users on SFA applications. This framework authenticates users based on biometric information. Once a user has been authenticated, they may attempt to access a protected resource within a web based application such as a link or button. The user will be authorized to view the protected resource only if the security role and security privileges match.

2.6.10 Servlet

Servlets accomplish a major application subsystem and divide it into one or more conversations. Each conversation consists of a set of request/response exchanges to complete a single activity. Each conversation in turn is divided into one or more commands that represent a single request/response exchange between the client and server. Together these servlets, conversations, and commands map to classes for conversation, exchange activity, and exchange presentations that represent the core of the servlet framework. The conversation class correlates commands with exchange activities and exchange presentations. The exchange activity class executes that business logic of the user requests. The exchange presentation class formulates and sends a response to the user.

The servlet framework defines and specifies usage of the following components:

- Conversation classes
- Exchange activity classes
- Exchange presentations classes

2.6.11 Session Management

The session management framework simplifies, standardizes, and extends the use of session information within the J2EE standard. The session wrapper class provides a common way to access session information, decouples session information from the request, session, and application J2EE contexts, and wraps WebSphere session extension classes. The session utility class provides utilities for sessions that are common across enterprise applications, such as detecting whether a browser accepts cookies.

The session class for large sessions provides a best practices and higher performance implementation for applications requiring storage of large amounts of session information. The session framework is integrated with the servlet framework for seamless use.



The session framework defines and specifies usage of the following components:

- Session wrapper class
- Session utility class
- Session class for large sessions

2.6.12 Timeout

The timeout framework supplements the basic WebSphere timeout services that allow for timeouts to be set for end-to-end transactions on a server level and transactions to JDBC datasources on a datasource level. The timeout framework is built to utilize a pool of timers. The usage of the pool allows applications to reuse timers and avoid repeatedly incurring the overhead associated with creating new threads each time an operation needs to be monitored for timeout.

The configuration of the timer pool is application independent and allows for multiple timeout intervals within an application. With this approach, an application can determine what is acceptable without having any knowledge of the application's timeout settings. The timer can also utilize the logging services and the exception handling methodology being implemented in the ITA R2.0 architecture.



3 SFA Projects

3.1 Overview

Based on existing documentation and interviews with application leads, this section of the strategic assessment documents the following information for each SFA project:

Section	Description
Functional Description	High level overview of the application's functionality.
Technical Description	Technical description of the application including volume of users, concurrent users, components (such as HTML and JSP pages), and products.
Status	Describes the current status of the application.
ITA 1.0 services used	Identifies the ITA 1.0 services used by the application.
Enhancements/Upgrades	Enhancements and/or upgrades required by the application.
Current/Potential ITA 2.0 services	Lists current ITA services and potential ITA 2.0 services that may be used by the application in its next release.
Schedule	Schedule of the application's development, testing, and production dates.

3.2 CBS2

3.2.1 Functional Description

The Campus-Based System 2 (CBS2) enables the Department of Education (ED) to provide more than \$2 billion in Title IV student financial assistance funds to approximately 4000 post-secondary institutions each year through a complex allocation model. The system provides allocations/authorizations for grant, work-study, and loan funds to these institutions, and the institutions, in turn, use these funds to provide student financial assistance to more than 1,000,000 needy students each year. There are several distinct programs, each with its own legislative history and regulations, within the Institution-Based Programs. These include the



Federal Supplemental Educational Opportunity Grant Program (FSEOG), Federal Work-Study Program (FWS), and Federal Perkins Loan Program (Perkins).

The CBS2 application provides the basis for funding/allocations and reallocation of funds to schools for needy students, the Fiscal Operations Report which summarizes the Institution-Based activity for a specific award year, and the Application to Participate process, which enables schools to request a continuance and/or an increase in their participation under the Institution-Based programs.

In summary, the CBS2 application calculates funding formulas, provides the basis for funding of schools for FSEOG, FWS, and Federal Perkins Loan programs. In addition, it supports the filing of the Fiscal Operations Report and Application to Participate (FISAP) via data received through EDConnect/SAIG.

As part of the overall Modernization program, the SFA sought to redesign the Institution-Based Programs' system to provide Institutions and staff with an integrated and intuitive solution that increases information access and self-service capabilities, therefore, supporting SFA's goals of reduced unit cost, customer satisfaction and employee satisfaction.

3.2.2 Technical Description

Volume	
Total Users	12,000
Concurrent Users	200
Components	
HTML pages	450
JSPs	60
Servlets	15
CGI	N/A
Other	180 Java Classes and Beans 30 Oracle Stored Procedures



ITA 2.0 Products
<ul style="list-style-type: none">• Visual Age for Java Enterprise 3.5• WebSphere Studio Advanced Edition 3.5• Sun Solaris Version 2.6• WebSphere Application Server 3.5• Java Server Pages (JSP) Specification 1.1• Java Development Kit (JDK) 1.2.2• Java Servlet Specification 2.1• Oracle 8i RDBMS v. 8.1.6• IBM eNetwork Dispatcher

3.2.3 Status

CBS2 is currently in development. There is no previous version of the application in production.

3.2.4 ITA 1.0 Services Used

There is no previous release of CBS2. Therefore, there is no basis of assessment.

3.2.5 Enhancements/Upgrades

The following upgrades have been performed in the CBS2 development environment:

- Upgraded to WAS 3.5
- Upgraded IHS
- Upgraded Oracle database
- Added Solaris patches to work with WAS 3.5

3.2.6 Current/Potential ITA 2.0 Services

CBS2 used the ITA 1.0 architecture selection and Java naming standards. ITA also set up SSL on the web server. CBS2 could use the following ITA 2.0 services:



- Logging Service
- Exception Handling Service
- Servlet Framework
- Persistence Framework
- Session Framework
- Component Factory
- Mail

CBS2 is planning on using the mass mailing service that has been developed by IFAP. This mailing service is comprised of three components including a sendmail utility program, an Oracle database, and a HP-UX Operating System (OS) scheduler. Although the service is in use, it has been difficult to debug and maintain because of the complexity and difficult integration of the three components.

CBS2 has contracted with an outside vendor for \$500k to conduct a code review, called PRR.

3.2.7 Schedule

Task	Start Date	End Date
<i>Phase 1</i>		
Development	3/15/01	6/19/01
Test	6/19/01	8/21/01
Production		9/15/01
<i>Phase 2</i>		
Development	5/11/01	10/31/01
Test	11/01/01	11/31/01
Production		12/01/01



3.3 CFO

3.3.1 Functional Description

The CFO Data Mart will enable the automation of the CFO's current budget planning, financial reporting, and activity based costing processes from ED's Financial Accounting and Reporting System (FARS). The CFO Data Mart will reduce this time consuming effort by automating the creation of the following monthly reports:

- SFA Plan vs. Actual Report
- SFA Current Plans and Fund Status Report
- SFA Operating Plan Report
- SFA Fund Balance Report
- Ad hoc queries, such as payroll by Channel, expenditures by project by Channel, etc.

The CFO Data Mart will provide a web interface to the data mart through SFA's intranet. CFO staff may use the application to conduct ad hoc queries and reporting (such as payroll by Channel, expenditures by project by Channel, etc.) on SFA accounting data in a flexible and efficient manner. The ad hoc reporting includes: analysis on existing reporting, customizing predefined reports, and creating new reports.

The Financial Management System (FMS) will replace FARS. FARS will remain the system of record until January 2002. The data mart will be built on FARS and continue to use it as the main source system for the next 18 months. Once FMS becomes the system of record the extract-transform-load (ETL) process will be modified to pull from FMS vice FARS and the transition will be seamless to the data mart users.

The CFO Data Mart will use Informatica, a COTS tool, to acquire data from the financial Management System (FMSS). Informatica will read the source data, and make the necessary transformations to populate the mart. User-supplied cross-reference data will also be used to populate the mart (e.g., Channel, Lim, Activity). FMSS will perform a nightly extract of activity from the previous day. This data will be exported via FTP to the Informatica server at the VDC. Informatica will then import the data into Oracle, perform the necessary transformations, and populate the CFO Data Mart. This data mart will be accessed by MicroStrategy's Intelligent Server to satisfy user requirements that are sent via a web interface.

The CFO Data Mart will use MicroStrategy software to provide the front-end user interfaces that allow reporting and On-line Analytical Processing (OLAP), and the back-end processing including SQL generation, load management, and report scheduling.



3.3.2 Technical Description

The Virtual Data Center (VDC) will house the Data Mart development and production database and production software servers.

Volume	
Total Users	6-8
Concurrent Users	6



Components	
HTML pages	N/A
JSPs	N/A
Servlets	N/A
CGI	N/A
Other	MicroStrategy COTS package for web interface
Products	
<ul style="list-style-type: none">• Informatica• MicroStrategy	

3.3.3 Status

CFO is currently in production. No development is currently planned.

3.3.4 ITA 1.0 Services Used

The CFO Data Mart uses the Data Warehousing products, MicroStrategy and Informatica, identified in ITA 1.0.

3.3.5 Enhancements/Upgrades

No enhancements or upgrades are planned for CFO.

3.3.6 Potential ITA 2.0 Services

No services have been requested since CFO is currently in production and no development has been scheduled.

3.3.7 Schedule

No future releases of CFO Data Mart are currently planned.



3.4 COD

3.4.1 Functional Description

The Common Origination and Disbursement (COD) process will enable SFA to deliver financial aid for Pell Grants, Direct Loans, and campus-based programs through a single process. A common process and system to support origination and disbursement will be critical if SFA hopes to achieve an enterprise-wide solution that will provide real-time data to students, schools, and financial partners via web portals. COD is a transaction-based payment method for schools. The team designed a “baseline” process that will be available for all schools. They also identified some areas in the process that could be used as “performance-type” indicators, thereby providing additional options and flexibility to schools that meet these measures.

3.4.2 Technical Description

Volume	
Total Users	20,000
Concurrent Users	1,500
Components	
HTML pages	N/A
JSPs	N/A
Servlets	N/A
CGI	N/A
Other	8 JavaBeans within the FTP framework
Products	
<ul style="list-style-type: none">• WAS 3.5• IHS 1.3.2• MQSeries 5.1 or 5.2• MQIntegrator 2.01	



3.4.3 Status

COD is currently in the design stage.

3.4.4 ITA 1.0 Services Used

COD did not participate in ITA R1.0.

3.4.5 Enhancements/Upgrades

No product upgrades.

3.4.6 Current/Potential ITA 2.0 Services

COD is planning on using the FTP common service.

3.4.7 Schedule

Task	Start Date	End Date
Common Record for Testing		5/31/01
Test	5/31/01	
Input Interfaces (CPS, PEPS)		8/31/01
Financial Interfaces (FMS, DLSS)		11/31/01
Production		2/19/02

3.5 DLM

3.5.1 Functional Description

The Delinquency Loan Mart (DLM) replaced the Central Data System (CDS) Data Mart. The DLM Data Mart provides SFA with the following reports and data files:

- The Department of Education Delinquency Report
- The Individual School Delinquency Report
- The Individual School Delinquency Data File



The initial release of the DLM leverages the existing Title IV wide area network (TIVWAN) architecture for report and file distribution. While the initial data mart release simply replaced the existing CDS delinquent loan reporting process, SFA may choose to leverage the advanced capabilities of the enterprise data warehouse architecture in future releases to enable ad hoc and trend analysis of delinquency data. Instead of simple, rigidly formatted reports, the data mart will enable SFA to deliver highly analytical reports to its customers. SFA will also have the option of delivering delinquent loan reports and analysis through a web interface.

The DLM will initially receive data from the Loan Servicing System and TIVWAN. Depending on future requirements, these data sources may change.



3.5.2 Technical Description

Volume	
Total Users	6-8
Concurrent Users	6
Components	
HTML pages	N/A
JSPs	N/A
Servlets	N/A
CGI	N/A
Other	Unix scripts/jobs run at the beginning of each month to produce delinquency reports on SL
Products ITA 1.0	
<ul style="list-style-type: none">• Informatica• MicroStrategy	

3.5.3 Status

DLM is in production.

3.5.4 ITA 1.0 Services Used

3.5.5 Enhancements/Upgrades

No enhancements or upgrades are planned.

3.5.6 Current/Potential ITA 2.0 Services

Since no future releases are planned, no ITA 2.0 services may be implemented.



3.5.7 Schedule

No future releases of DLM are currently planned.

3.6 EAI

3.6.1 Functional Description

The Enterprise Application Integration (EAI) project is an ITA R2.0 effort that will create an SFA EAI architecture for connecting front end systems to back end legacy systems via a reusable common architecture. EAI is a set of technology services that enables the sharing of processes and data of disparate systems to support end-to-end business processes. The EAI architecture enables the many “stovepipe” applications to exchange information via common, reusable methods and infrastructure. This application will provide the core messaging and sample transformation functionality to validate the EAI architecture and connectivity between the WebSphere Server and release one legacy systems.

The EAI will allow the SFA to integrate new web-based applications with existing back-end systems, while at the same time, providing a means to migrate away from reliance upon existing legacy systems.

EAI provides capabilities that will allow for the integration of web-based applications, the Data Warehouse environment, commercial-off-the-shelf (COTS) packages, and existing legacy systems within the SFA technical environment.

The first release of the EAI application will provide legacy connectivity to the COD application and the following back end legacy applications: CPS, DLSS, NSLDS, PEPS, TIVWAN/bTrade. The EAI application will provide messaging standards, implement guaranteed message delivery, clustering, and failover. MQIntegrator will be used to validate non-application specific message transformation and routing. This effort will include the build, test, deploy of the EAI application and provide technical specifications for the architecture.

The EAI architecture provides the following technical services:

- Communications Middleware
- Transformation and Formatting
- Application Connectivity
- Business Process Management



3.6.2 Technical Description

Volume	
Total Users	not yet available
Concurrent Users	not yet available
Components	
HTML pages	not yet available
JSPs	not yet available
Servlets	not yet available
CGI	not yet available
Other	MQSeries adapters, etc.
Products	
<ul style="list-style-type: none">• WAS 3.5• IHS 1.3.2• MQSeries 5.1 or 5.2• MQIntegrator 2.01	

3.6.3 Status

EAI is currently in development.

3.6.4 ITA 1.0 Services Used

3.6.5 Enhancements/Upgrades

The team is not sure how they will be able to maintain their development environment as they move through development. Due to the fact that they only have two boxes for the EAI application, they are planning on moving both boxes into system test which would not give them a development environment to fix bugs. Furthermore, when the team moves into production, they are planning on moving these system test boxes into production. As a result,



they will not have a development environment to fix the code, nor will they have a system test environment to regression test the application.

No enhancements or upgrades are planned.

3.6.6 Current/Potential ITA 2.0 Services

Since the EAI code is expected to be completed by 5/31/01, ITA will not be able to deliver any common services in time for EAI to use in this release.

3.6.7 Schedule

TBD.

3.7 EIP

3.7.1 Functional Description

The Enterprise Information Portal (EIP) is a web-based enterprise wide portal, with separate views based on the audience. The application will consist of five separate views catering to the audience, namely Students, Schools, Financial Partners, Employees of SFA, and a public view for the general Internet audience. These views will contain information pertaining to their respective channel, and will allow for a seamless and transparent distribution of pertinent information from a variety of back-end data sources, and assist the users in their daily work.

The EIP effort is a replatforming of the existing Schools Portal application.

3.7.2 Technical Description

Volume	
Total Users	50,000
Concurrent Users	2,000
Components	
HTML pages	40
JSPs	30
Servlets	25



CGI	configurable with templates from Autonomy
Other	



Products
<ul style="list-style-type: none">• Visual Age for Java Enterprise 3.5• WebSphere Studio Advanced Edition 3.5• Sun Solaris Version 2.6• WebSphere Application Server 3.5• Java Server Pages (JSP) Specification 1.1• Java Development Kit (JDK) 1.2.2• Java Servlet Specification 2.1• Oracle 8.1.6• IBM eNetwork Dispatcher

3.7.3 Status

EIP is currently in development.

3.7.4 ITA 1.0 Services Used

EIP did not participate in ITA R1.0.

3.7.5 Enhancements/Upgrades

EIP will implement the new version of Autonomy.

3.7.6 Current/Potential ITA 2.0 Services

EIP will use the standard ITA R2.0 product suite and the following common services:

- Logging
- Exception Handling
- Search
- Mail





3.7.7 Schedule

Task	Start Date	End Date
Analysis	4/06/01	4/20/01
Build	4/20/01	5/01/01
System Test	5/01/01	6/15/01
Performance Test	6/15/01	7/15/01
Production/Go Live	7/31/01	7/31/01

3.8 FAFSA

3.8.1 Functional Description

Free Application for Federal Student Aid (FAFSA) on the Web is an application used by college students and schools to submit financial applications via the Internet. During the 2000-2001 academic year, over two million students will use FAFSA on the Web to apply for Student Financial Aid. SFA expects the volume to double during the 2001-2002 academic year and to continue to rise in future years.

The application replaced a paper form and previous version located at fafsa.ed.gov. The online application includes approximately 80 questions that are used to determine students' eligibility for financial assistance. Students can create, review, renew, or correct forms by accessing the site with Personal Identification Number (PIN).

The FAFSA on the Web modernization initiative is an effort to modernize FAFSA on the Web while accommodating the anticipated increase in users. The initiative will improve customer satisfaction by providing students who use the site a complete user experience in applying for Federal, State, and Institutional aid. Customer satisfaction will be improved by increasing the application's usability, accuracy, efficiency, security, and supportability/scalability. SFA will also benefit from the reduction of costs due to reduced manual processing of applications and fewer customer service calls as a direct result of an application that is easier to use.

Usability will include all day everyday access, a real-time display of results from the Central Processing System (CPS), reusable data that can be shared between different applications, and the incorporation of the latest best practices in web user interface design.



3.8.2 Technical Description

Volume	
Total Users	2 – 4 million
Concurrent Users	1,000 – 2,000
Components	
HTML pages	245
JSPs	163
Servlets	10
CGI	N/A
Other	40 Java Classes
ITA 2.0 Products	
<p>The development and test environments use:</p> <ul style="list-style-type: none">• Visual Age for Java Enterprise 3.5• WebSphere Studio Advanced Edition 3.5• Sun Solaris Version 2.6• WebSphere Application Server 3.5• Java Server Pages (JSP) Specification 1.1• Java Development Kit (JDK) 1.2.2• Java Servlet Specification 2.1• Enterprise JavaBean Specification 1.0• Oracle 8.1.6• IBM eNetwork Dispatcher	



3.8.3 Status

FAFSA release 5.x is currently in test. The previous version is in production.

3.8.4 ITA 1.0 Services Used

ITA R1.0 has provide FAFSA with the following support:

- Setup the development and test environments
- Designed performance test environment
- Provided architecture reviews
- Setup SSL and common connection pooling

3.8.5 Enhancements/Upgrades

FAFSA will require the following enhancements/upgrades:

- Build an ITA 2.0 performance test environment for FAFSA
- Use eNetwork Dispatcher for intelligent load balancing
- Upgrade system test environment to WAS v. 3.5
- FAFSA is considering moving from Shadow Direct to an EAI (MQ Integrator) solution for Release 6.0

3.8.6 Current/Potential ITA 2.0 Services

FAFSA will use the ITA 2.0 product suite.

Future FAFSA releases may be able use the following common services:

- Logging
- Exception Handling
- Performance Monitoring - utilization per application, CPU and memory usage (Implementing the performance monitoring service would help FAFSA with its capacity planning.)
- Provide best practices, reviews, connection pooling, session management, use of EJBs



3.8.7 Schedule

The current release of FAFSA is Web redesign 5.x

Task	Start Date	End Date
5.x schedule		
Design	8/28/00	2/2/01
Build	2/5/01	4/27/01
System Test	3/21/01	6/15/01
Performance Test	4/23/01	6/15/01
Deliver - fully tested version of the first release of the FAFSA redesign code transitioned to NCS. NCS will use the code as a baseline for Release 6.0, scheduled for production January 1, 2002.	6/29/01	6/29/01

3.9 FP Data Mart

3.9.1 Functional Description

The Financial Partner's (FP) Data Mart will provide executive information and decision support capabilities around several key business functions for both the Guaranty Agencies (GAs) and the Lenders. The scope for the Data Mart initiative will establish the Data Mart infrastructure within the Channel and will provide initial Risk Management, Customer Relationship Management, Compliance Management, and Portfolio Management functionality related to the thirty-six Guaranty Agencies and the approximately four thousand lenders.

The FP Channel is responsible for both current and retired loan programs and, as such, is under customer and program obligation to ensure the capture and comparative ability of all programs. By collecting information (associated with student lending among Financial Partners including Guaranty Agencies, Lenders, and Servicers particularly in the FFEL Student Loan Program) from several sources into a central location, personnel in the Channel as well as



external partners will be able to more efficiently identify areas in which each party may assist the other while improving the support for students within the Federal Family Education Loan Program.

The FP data mart receives data from the National Student Loan Data System (NSLDS), The Federal Family Education Loan System (FFEL), The Post-secondary Education Participants System (PEPS), and FMS. Using this information, the FP data mart performs reporting and analysis on loan level data and Individual Lender and Guarantee Agencies. The FP data mart will include Analysis, Decision Making, Trend, and Risk Assessment Requirements reporting. Various reporting (i.e.: the relationships between Guaranty Agencies, Lenders, and Servicers; identification of 'top' Guaranty Agencies and Lenders; the risk scorecard) will be developed to support some of the analytic and decision support processing that is required. Analytic and decision support reporting includes 'what-if' analysis, trending, and benchmarking in areas related to the operations, default management, loan servicing, financial information, and regulatory requirements of the Guaranty Agencies and Lenders.

The Data Mart will be implemented in three releases in order to populate the Data Mart with information from the identified four source systems. Release 1 will establish the Data Mart infrastructure within the Channel and populate the Data Mart with Guaranty Agency and Lender information from the FFEL system. Release 2 will populate the Data Mart with loan level information from NSLDS. Release 3 will populate the Data Mart with current Guaranty Agency and Lender reports from FMS and with information related to performance reviews and audits from PEPS.

3.9.2 Technical Description

Volume	
Total Users	36 GAs 4,000 lenders
Concurrent Users	20
Components	
HTML pages	10
JSPs	N/A
Servlets	N/A



CGI	N/A
Other	N/A



Products
<ul style="list-style-type: none">• MicroStrategy 7.0.2 Web Server & Intelligent Server• MS IIS Web Server• Informatica• Oracle 8i• Network Dispatcher (as load balancer)

3.9.3 Status

FP Data Mart finished its analysis phase and has started development.

3.9.4 ITA 1.0 Services Used

ITA 1.0 setup the FP Data Mart development and test environments. The product vendors provided software setup and product support. MicroStrategy and Informatica consultants will perform the development and test and they have their own DBA

3.9.5 Enhancements/Upgrades

FP Data Mart is using the current product installations. The FP Data Mart is planning to upgrade to MicroStrategy 7.1 Web & Intelligent Server and Oracle 8.1.6. (The FP Data Mart DBA will perform the Oracle upgrade.) Servicepack may also need to be added to the MS software.

3.9.6 Current/Potential ITA 2.0 Services

FP Data Mart may be able to use the FTP framework.

FP Data Mart is planning on 8 hours per week of ITA support and 72 hours for MicroStrategy and Informatica support.

3.9.7 Schedule

A multiple release approach will be used to implement the initial set of functionality for the Data Mart. The Data Mart will be divided into three releases.



Task	Start Date	End Date
<i>Release 1</i>		
Analysis	2/15/01	3/26/01
Development	3/27/01	4/22/01
Performance Test	4/23/01	5/11/01
Production/Go Live	5/11/01	5/21/01

3.10 IFAP

3.10.1 Functional Description

The Information for Financial Aid Professionals (IFAP) web site (<http://ifap.ed.gov>) is an electronic library for financial aid professionals containing publications, regulations, and guidance regarding the administration of the Title IV Federal Student Financial Aid Programs. IFAP is managed by the Customer Support Branch (CSB) within the Office of Student Financial Assistance Programs (OSFAP) of the Department of Education.

The IFAP web collection went “live” in early March 1998. IFAP consists of forty-one (41) specific types of publications that CSB posts.

IFAP has a subscription option that notifies registered customers via email when new information (Dear Partner Letter, Announcement, Federal Register, etc.) has been added to the wealth of information available on IFAP. The email is customized based on users’ subscription selections.

Users can select to be notified when new documents are added to the IFAP catalog based on Publication Type and Program/Service categories. Users can select more than one Publication Type or Program/Service category. Even if selections overlap, the system is designed to not send duplicate notifications of a single document.

The IFAP subscription option sends “Weekly” (a weekly summary of all IFAP additions) subscription emails and “48 Hour” (a summary of IFAP additions within 48 hours) emails based on the end user’s subscription options form. Users must be registered to use the IFAP subscription option.



3.10.2 Technical Description

Volume	
Total Users	4000
Concurrent Users	40
Components	
HTML pages	100
JSPs	230 (with very little Java in the pages)
Servlets	N/A
CGI	N/A
Other	9,550 static documents
Products	
<ul style="list-style-type: none">• WebSphere Application Server Advanced Edition 3.02• IBM HTTP Server 1.3.6• Autonomy Knowledge Suite 2.1• Oracle 8.1.5• IBM eNetwork Dispatcher• Interwoven	

3.10.3 Status

IFAP is in production. There are no future releases planned. Bug fixes are ongoing.

3.10.4 ITA 1.0 Services Used

IFAP used several ITA 1.0 services. Specifically, ITA built the development, test, staging, and production environments. ITA also migrated product and code between the development, test, staging, and production environments. ITA also provided technical SME support for IHS,



WAS, and Autonomy. ITA also provided environment support throughout the application's testing phase.

3.10.5 Enhancements/Upgrades

IFAP may require an upgrade to WAS 3.5 and for Interwoven. This upgrade would also require a reinstallation of agents on the four production servers. A patch may also be required to keep ND outages from occurring.

3.10.6 Current/Potential ITA 2.0 Services

If IFAP produces another release of application code, the following ITA 2.0 common services could be implemented:

- ND Clustering
- Database framework
- Exception Handling (IFAP currently uses only one default error page, there is no way to track exceptions that have occurred historically.)

3.10.7 Schedule

IFAP is currently in production. No new versions are planned within ITA R2.0.

3.11 Intranet (SFANet) 2.0

3.11.1 Functional Description

SFANet Release 2.0 consists of approximately 300 pages of searchable content. The content is grouped by SFA Channels, Business Units, and Enterprise Services, with additional categories for News, Reference, and Forms. SFANet can be accessed by anyone with an EDLAN account using Microsoft Internet Explorer 4.0 or higher. All content on SFANet is printable.

The content on SFANet is managed through Interwoven TeamSite software. Content changes and updates are made to a virtual site that is stored on the team site development server. Once changes have been made to the virtual site, the edited site is migrated to production through the Interwoven Open Deploy module.

SFANet currently provides access to the SFA Stars Nomination page and the Feedback form.

The SFA Stars nomination form includes fields that allow users to nominate a SFA employee as a "Star," recognizing him or her for a job well done. The nominator enters his or her name, the



name of the nominee and a brief narrative to support the nomination. Upon submitting the nomination, the application sends emails to the nominator, nominee, and Human Resources.

The feedback form allows users to send comments to SFA Communications and the Intranet Application Management Team. The form can be accessed through a button on the frame of any SFANet page. The form has input fields for the feedback provider's name, email address and comments. A "Submit" button sends feedback in the form of an email message to the Webmaster and the Content Manager.

Both applications are servlets and composed by an HTML page that resides on the Team Site server and a class file that resides on the WebSphere application server.

SFANet uses the Autonomy search engine. This search engine can process keyword queries, Boolean queries or natural language queries. The primary search interface is a small search field located in the upper right corner of the frame found on all SFANet pages. A radio button selector allows users to choose between searching the entire site and searching the current subsite only. The user can type a search query in the search box and hit a "GO" button next to the box. The search engine will then return a page of query results. The search engine searches the actual content of all files and documents within the site. Each search result is displayed with a weight, which is a percentage based on the probability of a match to the particular query. Results are displayed in order of decreasing weights.

3.11.2 Technical Description

The Autonomy application resides on SU35E2 (dev/test/stag) and SU35E7 and SU35E14 (production).

Volume	
Total Users	1200 potential users
Concurrent Users	200
Other	
Components	
HTML pages	500
JSPs	0
Servlets	2
CGI	Makes use of CGI,



	derived functionality from Autonomy APIs
Other	Giga Content



Products
<ul style="list-style-type: none">• WebSphere Application Server Advanced Edition 3.02• IBM HTTP Server 1.3.6• Autonomy Knowledge Suite 2.1• IBM eNetwork Dispatcher• Interwoven

3.11.3 Status

The Intranet 2.0 is in production. Version 2.+ is in development, but the schedule has not been released.

3.11.4 ITA 1.0 Services Used.

In addition to the ITA core products listed above, the Intranet used several ITA 1.0 services. Specifically, ITA built the development, test, staging, and production environments. They also migrated product and code between the staging, and production environments. ITA also provided technical SME support for IHS, WAS, Autonomy, and Interwoven. ITA also helped troubleshoot the Mail APIs.

3.11.5 Enhancements/Upgrades

The Intranet may need to upgrade to WAS 3.5, IHS 1.3.12, and Interwoven. Network Dispatcher (ND) may also need to be upgraded (project wide) to fix ND synchronization of heartbeats. However, autonomy does not require an upgrade.

3.11.6 Current/Potential ITA 2.0 Services

The Intranet would benefit from utilizing the ITA coding standards for SQL and Java and the Exception Handling and Logging common services. The application would also benefit from implementing a common look and feel.

The Intranet currently uses a mail-based application that uses a database. This mail-based application is problematic and has multiple points of failure.



3.11.7 Schedule

TBD.

3.12 Schools Portal

3.12.1 Functional Description

The schools portal provides a central aggregation point for schools. The portal enables data transmissions and real-time inquiries, and provides schools with a centralized data repository of news, updates, deadline reminders, regulatory information, access to GAPS, “Dear Colleague” letters, and more.

3.12.2 Technical Description

Volume	
Total Users	50,000
Concurrent Users	2,000
Components	
HTML pages	40
JSPs	30
Servlets	25
CGI	configurable with templates from Autonomy
Other	
Products	
<ul style="list-style-type: none">• Autonomy• I.H.S v. 1.36• Viador• JRun• Interwoven	



3.12.3 Status

The Schools Portal is in production. There are no future releases planned. Bug fixes are ongoing.

3.12.4 ITA 1.0 Services Used

The Schools Portal used several ITA 1.0 services. Specifically, ITA built the development, test, staging, and production environments. They also migrated product and code between the development, test, staging, and production environments. ITA also provided technical SME support for HIS, Viador, Autonomy, JRun, and Interwoven. ITA also provided a workaround for a database issue with JRun and installed Viador patches to correct a database problem.

3.12.5 Enhancements/Upgrades

No enhancements or upgrades are currently planned for the Schools Portal.

3.12.6 Current/Potential ITA 2.0 Services

No new development is scheduled for the Schools Portal. Because the Schools Portal did not use WAS none of the common services would be available for future development.

3.12.7 Schedule

The Schools Portal is currently in production. There is no new version planned within ITA 2.0. The Enterprise Portal will probably replace the application.



4 Assessment Summary

The assessment summary section provides information on the strengths and enhancement areas of the ITA R1.0 architecture. This section also includes recommendations to address the enhancement areas identified in ITA R1.0. The enhancement area, recommendation, and priority level are listed in the recommendation matrix.

4.1 Strengths

Application teams identified the standardized architecture and technical support (including environment, operations, SME, and bug fixes) as the primary strengths of the ITA R1.0 architecture.

4.1.1 Standard Product Framework

ITA R1.0 provided a standard architecture framework for the applications. The standardized framework provided application teams with a robust framework that was scalable, configurable, and extendible. The standard ITA R1.0 and R2.0 products are listed in Table 1: ITA Products and Versions. For the most part, applications used elements of the ITA R1.0 architecture.

4.1.2 Operations Support

The ITA team provided the application teams with high-quality and on-time support.

4.1.3 Build/setup Environments

Because of the standardized ITA framework, the ITA team provided a smooth process for building application environments. Based on the application teams' proposed configurations and business requirements, the ITA team built development and test environments.

4.1.4 SME Support

The ITA team provided SME support for the ITA R1.0 core products including WAS, IHS, and Autonomy.

4.1.5 Troubleshooting and bug fixes

The ITA team assisted application teams debug and resolve problems identified throughout the development and testing cycles. The ITA team also provided some Java and JavaScript



debugging and programming assistance to Production Operations to identify and resolve problems in the Intranet, IFAP, and Schools Portals applications.

4.2 Enhancement Areas

The key enhancement areas identified for ITA R1.0 are communication, lack of using the ITA framework and core services, and a need for additional ITA services.

4.2.1 Communication

Application teams identified communication with ITA as a primary concern. Specifically, during the implementation of ITA R1.0, there was no clearly defined process for communicating with or submitting issues to the ITA team. Additionally, application teams did not have an ITA representative/liaison on their teams. Application teams were not consistently notified about scheduled outages. Because of the lack of a formalized communication structure with the ITA and application teams, some application developers sought ITA help too quickly, prior to working out their own bugs.

4.2.2 Use of ITA core services

One of the main weaknesses of the ITA R1.0 architecture was application teams not utilizing the core services provided by ITA. During the first release, the application teams did not plan for ITA support and therefore failed to use potential services including technical architecture design reviews and capacity planning.

4.2.3 Production Hardware Implementation

Some of the applications do not provide failover or load balancing in production because the applications are hosted on one server. These production configurations were based on business decisions, however, it should be noted that these features are potential weaknesses in the applications.

4.2.4 Issue Tracking

Application teams did not use an issue tracking or System Investigation Report (SIR) process during ITA R1.0. The lack of a standard, formalized issues tracking process led to problems communicating issues to ITA. For example, new bugs would be entered and assigned in a spreadsheet and no one would inform the ITA team that they had been assigned bugs. Furthermore, bug fixes that were made could not be clearly communicated and tracked on a spreadsheet.



4.2.5 Configuration Management

None of the application teams used configuration management in ITA R1.0. ITA was responsible for the manual code migration of the applications. For static content, the content management tool was used in production for IFAP and in development and production for the Intranet.

4.2.6 ITA R1.0 services too general

One of the application teams (FAFSA) noted that the ITA R1.0 framework and services were too general and did not provide many of the things that they needed. As a result, the application could not accurately plan and place the ITA services in their work plan and they had to pioneer many technical architecture services on their own. In addition, the ITA 1.0 security deliverable did not provide the application team with quantifiable technical advantages.

4.2.7 Exception Handling

ITA R1.0 did not provide the application teams with a common method for catching and handling exceptions. As a result, errors were not dealt with consistently which caused outages for several applications.

4.2.8 Logging

Few applications made use of Java logging in ITA 1.0. Furthermore, there was no standard way of logging in ITA R1.0. Applications that did use logging had no way to turn their logging levels "down" or off in production which may have caused logs to fill disks and bring the applications down. Additionally, logging was rarely used to produce error messages which could assist the operations team in fixing production bugs.

4.2.9 Database

Some applications overlooked database considerations such as connection pooling. Non-scheduled database outages hung the Schools Portal application. ITA developed a workaround for the scheduled database outages which requires a manual restart of the application server.

4.2.10 Mail

ITA R1.0 did not provide a common method for applications to send mail. As a result, the applications implemented mail functionality differently. These different mail functionalities also contain varying issues. The Intranet's mail functionality can be improved for performance and error recoverability. Users must wait while mail is being sent. Since the API does not start



a new thread to improve the performance speed. IFAP's mail capability is error prone, complex, and hard to debug.

4.2.11 Process Monitoring

CA Unicenter's TNG does not monitor production processes that need to be running. On several occasions servers and/or server processes have gone down for several days without the VDC knowing and resolving the issue.

4.2.12 Lack of Software Funding

The ITA R1.0 had a slight shortage of funds for its production environment. This was particularly apparent with Schools Portal which is only running on a single box that has no load balancing and no failover. Originally the funding was secured for a second server which would have provided the load balancing but not enough funding was secured to procure a second license for Viador/JRUN on the second server.

4.2.13 Security

The ITA did not provide SFA applications with clear security guidelines regarding session management and authentication. For example a user would be able to login to the IFAP application based on user id and password but they were not able to logout. This meant that a user would either have to shutdown their browser or wait for their user session to timeout.



4.3 Prioritized Recommendations

The following table summarizes the prioritized recommendations for ITA R2.0. The table provides a priority (High, Medium, or Low), enhancement area, and recommendation for the enhancement areas identified for ITA R1.0.

Priority	Enhancement Area	Recommendation
Medium	Communication	Formalize communication process between ITA and application teams. Implement SLAs to establish roles and response times for tasks.
High	Failure to use ITA core services	Conduct technical architecture design reviews.
High	Production Hardware Implementation (lack of capacity planning, load balancing, and failover)	Conduct technical architecture design reviews to ensure that applications are following ITA recommendations for production implementation.
Medium	Issue Tracking	Implement enterprise-standard issue tracking tool and process.
Medium	Configuration Management	Implement enterprise-wide configuration management tool.
Medium	ITA R1.0 services too general	Provide new set of common services to application teams.
High	Exception Handling	Implement Exception Handling Framework.
High	Error Logging	Implement Logging Framework
Medium	Database	Distribute developer's handbook and best practices guides.
Medium	Mail	Implement Mail framework.
Low	Process Monitoring	Upgrade NetworkDispatcher.
Medium	Lack of software funding	Prioritize software funding.
Medium	Security	Implement security framework.



Table 2: Prioritized Recommendations for ITA R2.0



4.3.1 Common Services

Based on the application requirements, the ITA team has identified the following priority for developing and deploying the ITA R2.0 common services:

Priority	Service
1	Exception Handling
2	Logging
3	Mail
4	FTP
5	Servlets
6	Session
7	Persistence
8	Component Factory
9	Security
10	Timeout
11	Configuration
12	Search



Appendix A – Application Common Services Matrix

See attached spreadsheet.



Appendix B – Integrated Technical Architecture Product Standards

See attached table.